INFORMATION AND CONTROL **70**, 1–31 (1986)

Functional Dependencies and Constraints on Null Values in Database Relations*

PAOLO ATZENI

Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185 Roma, Italy

AND

NICOLA M. MORFUNI†

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," Via Buonarroti 12, 00185 Roma, Italy

Database relations with incomplete information are considered. The no-information interpretation of null values is adopted, due to its characteristics of generality and naturalness. Coherently with the framework and its motivation, two meaningful classes of integrity constraints are studied: (a) functional dependencies, which have been widely investigated in the classical relational theory and (b) constraints on null values, which control the presence of nulls in the relations. Specifically, three types of constraints on null values are taken into account (null-free subschemes, existence constraints, disjunctive existence constraints), and the interaction of each of them with functional dependencies is studied. In each of the three cases, the inference problem is solved, the complexity of the algorithms for its solution analyzed, and the existence of a complete axiomatization discussed.

1.0. INTRODUCTION AND MOTIVATION

Database systems usually handle large quantities of highly structured data. The relational model, which is the most popular in the research community, and is becoming widely available in the commercial world, allows data to be organized as sets of n-tuples of fixed format. On the other hand,

* A preliminary version of this paper was presented at the 11th International Colloquium on Automata, Languages, and Programming (ICALP), in Antwerp, Belgium, July 1984, under the title "Functional Dependencies and Disjunctive Existence Constraints in Database Relations with Null Values."

† Partially supported by a grant from Data Base Informatica SpA, Italy. Current affiliation: SARIN, SpA, S.S.148 Pontina, Km 29.100, 00040 Pomezia, Italy.

0019-9958/86 $3.00

these data have to represent information on a fragment of the real world of interest for the system, and the available information need not fit into the given format. The easiest way to deal with these situations is to use special values (called *null values* or, simply, *nulls*) to indicate lack of complete information. Nulls have been studied by various authors (see Zaniolo, 1984, or Maier, 1983, for a review), in order to extend the theory of relations (which, usually, assumes, for the sake of simplicity, complete information) to the new framework. Various interpretations have been proposed for the nulls and some interesting properties have been shown for each, but no complete theory has been formulated for any of them. Among the various proposals, the "*no-information*" interpretation (Zaniolo, 1984), under which a null associated with an attribute in a tuple means that no information is available about that attribute for that tuple, is very interesting: it is the most primitive, but, at the same time, it can be used to model every kind of missing or incomplete information, and its semantics is certainly simple and well understood.

The use of null values in database relations allows to deal with incomplete information, increasing the capability and flexibility of relations in capturing the semantics of the reality of interest. Consider the relation scheme R (Project, Part, Supplier) representing the relationship between parts, their suppliers, and the projects that use them. It may often be the case that incomplete information is available: for instance, it may be known that a supplier can supply a part or that a project uses a part. So relations with null values (see an example in Fig. 1) represent typical real-world situations. According to the no-information interpretation associated with null values, no supposition is made about the attribute *project* in the second tuple or about the attribute *supplier* in the third one (a project using part "p_2" supplied by "s_2" may either exist but be at present unknown or not exist; analogously for the supplier of "p_3").

One of the most important areas of research in database theory is that of integrity constraints, which are properties that must be satisfied by the relations in the database. They are used to represent semantic properties of the real world that cannot be captured by the flat structure of relations, and play a crucial role in the design theory of relational databases: their

Project	Part	Supplier
j_1	p_1	s_1
$\varnothing$	p_2	s_2
j_1	p_3	$\varnothing$

FIGURE 1

knowledge is used in order to design "better" relation schemes (they form the basis of the so called "database normalization theory"; see, for example, Ullman, 1982, or Maier, 1983); as a consequence, their properties have been deeply investigated with regard to database relations without null values. In such a framework, *functional dependencies* (FDs) are the most natural and, as a consequence, studied class of integrity constraints: they represent functional relationships between classes of objects in the real world. Recently, various authors (Vassiliou, 1980, Lien, 1982, Imielinski and Lipski, 1983) have considered FDs with regard to relations with null values (abbreviated NFDs). Only the treatment in Lien (1982) is suitable of application to null values under the no-information interpretation and we will therefore adopt it.

The use of nulls allows the possibility of representing a larger class of situations; as a consequence, this requires new classes of integrity constraints to be introduced to represent new semantic properties of the real world and of our knowledge of it. Let us consider an example. In a relation over the attributes *Social Security number* (SSNo), *category*, *age*, *salary*, it would be meaningless to have nulls for the attribute SSNo, since its values are the only means of accessing the tuples. It is interesting to remind that in Codd (1970), where the relational model was first proposed and the possibility of using null values briefly examined, similar considerations were developed about the attributes in the primary key.

In other situations, the presence or absence of null values for some attributes may be related to their presence or absence for some other attributes. For example, in a relation over the attributes *department*, *manager's last name*, *manager's first name*, a first name should be present only if the corresponding last name is present. An even more complex situation may be the following: in a relation with scheme SSNo, *first name*, *last name*, *birth date*, *birth place*, and other attributes, it may be required that if the last name is not null, then the SSNo or all of first name, birth date and birth place are not null. In order to deal with the situations exemplified above, three kinds of constructs have been introduced: "*null-free subschemes*" (Atzeni and Morfuni, 1984a), "*existence constraints*" (Maier, 1980), "*disjunctive existence constraints*" (Maier, 1980, Goldstein, 1981), which we will refer to with the generic phrase *constraints on null values*.

The aim of this paper is the study of the interaction between functional dependencies and constraints on null values. In particular, we address the implication problem (the problem of deciding whether a constraint c holds in all the relations that satisfy a set of constraints C) and its solution by means of systems of inference rules. This problem is crucial in all the situations that require the use of integrity constraints in the design process (see, e.g., Maier, 1983) and so its solution is particularly important. In presence of null values (and of constraints on them), implication of constraints may be applied to decomposition of relations from one further point of view, beside the usual ones: decompose a relation with nulls into relations where the presence of nulls is somehow minimized. The implication problem for functional dependencies in relations without nulls was studied in Armstrong (1974), with efficient algorithms presented in Beeri and Bernstein (1979), while inference rules for functional dependencies with null values, for existence constraints, and for disjunctive existence constraints were presented in Lien (1982), Maier (1980), and Goldstein (1981), respectively. Some results on the interaction between functional dependencies and constraints on null values were presented in Atzeni and Morfuni (1984a, b), which are completed by the results presented here.

The paper is organized as follows. Section 2 contains the reference background, while Sections 3, 4, 5, present the study of the interaction between functional dependencies and, respectively, null-free subschemes, existence constraints, and disjunctive existence constraints.

2.0. Background

2.1. *Basic Concepts*

In relational theory, *attributes* are symbols taken from a finite set U. In the following we use the first letters of the alphabet, $A, B, C, \ldots$, possibly with subscripts, to indicate single attributes and the last letters $Z, Y, X, \ldots$, to indicate sets of attributes. Moreover, we write A to indicate both a single attribute A and the set $\{A\}$, and XY to indicate the union of two sets X and Y. With each attribute A, there is associated a set of values, $\mathrm{Dom}(A)$, called the *domain* of A.

In the classical theory (i.e., that referring to relations without nulls), a *tuple* over a set of attributes U is a mapping t which associates a value of $\mathrm{Dom}(A)$ with each attribute $A \in U$. The value associated with an attribute A is indicated with $t \cdot A$; the same notation is generalized to sets of attributes: $t \cdot Y$ (with $Y \subseteq U$) indicates the restriction of the mapping t to the attributes in Y.

A *relation* r over a set of attributes U is a set of tuples on U. The time invariant structure of a time varying relation r over the attributes U is indicated with $R(U)$ and called *relation scheme.*

In order to allow the presence of nulls we modify the definition of tuple: a tuple over a set of attributes U is a mapping t that associates with each attribute $A \in U$ either a value of $\mathrm{Dom}(A)$ or the null value $\varnothing$. A tuple t is *total on A* (or *A-total*) if $t \cdot A$ is not null and *total on X* (*X-total*) if it is total on each attribute $A \in X$. If a tuple is U-total (i.e., total on all the attributes and so null-free) it is said *total.* Again, a relation r over the scheme $R(U)$ is a set of tuples on U.

To give a formal description of relations containing no-information nulls, we can reason as follows. Given a relation over a scheme $R(U)$, we have, for each non-empty subset X of U, one $|X|$-ary predicate symbol P_X and (if X is not a singleton) an implication statement of the form $\forall a, y$ $(P_X(a, y) \Rightarrow P_Y(y))$, for each proper subset Y of X with cardinality $|X| - 1$. Then, for each tuple t in the relation, if X is the maximum subset of U such that t is X-total, we have an atomic sentence $P_X(x)$, where $t \cdot X = x$. In the example in Fig. 1, we would have, at the scheme level, seven predicate symbols, P_{JPS}, P_{JP},..., P_S, and nine implication statements:

$$\forall j \, \forall p \, \forall s \, (P_{JPS}(j, p, s) \Rightarrow P_{JP}(j, p))$$

$$\forall j \, \forall p \, \forall s \, (P_{JPS}(j, p, s) \Rightarrow P_{JS}(j, s))$$

$$\ldots$$

$$\forall p \, \forall s \quad (P_{PS}(p, s) \Rightarrow P_S(s)),$$

and, at the instance level, the three sentences $P_{JPS}(j_1, p_1, s_1)$, $P_{PS}(p_2, s_2)$, $P_{JP}(j_1, p_3)$.

2.2. *Functional Dependencies*

In the classical theory, a *functional dependency* (FD) is a statement $f: X \rightarrow Y$, where X, Y are sets of attributes. A null-free relation r over a scheme $R(U)$ (with $XY \subseteq U$) *satisfies* f (we say also that f *holds* in r) when, for each pair of tuples $t_1, t_2 \in r$, if $t_1 \cdot X = t_2 \cdot X$ then $t_1 \cdot Y = t_2 \cdot Y$.

Useful concepts in dependency theory are those of implication and inference rule. Given a set of constraints that hold in a relation it is often possible to deduce that other constraints also hold in that relation. A constraint i is *implied* by a set of constraints I on a relation scheme $R(U)$ if it holds in all the relations that satisfy all the constraints in I. The set of all the constraints implied by I is called the *closure* of I and indicated with I^+. Two sets of constraints are *equivalent* if their closures are identical. Given I and i, the *implication* (or *membership*) *problem* is to tell whether I implies i. Most of the decision problems for dependencies can be reduced to the implication problem and so its efficient solution is a prerequisite to the use of dependencies in the design process. The algorithms for its solution (called *membership algorithms*) have correctness proofs that are usually based on sound and complete sets of inference rules. An *inference rule* is a rule that allows the derivation of a constraint from some other constraints. The basic requirement for each inference rule is to be *sound*, that is, to derive from I only constraints that are in I^+. Moreover, it is important to have sets of inference rules that are *complete*, i.e., that allow the derivation of all the constraints in I^+.

It is well known that for FDs in the relational model without nulls the following is a sound and complete set of inference rules (the first sound and complete set was presented in Armstrong, 1974, while this is from Ullman, 1982):

F_1 (reflexivity). If $Y \subseteq X$, then $X \rightarrow Y$ holds

F_2 (augmentation). If $X \rightarrow Y$ holds, then $XZ \rightarrow YZ$ also holds

F_3 (transitivity). If $X \rightarrow Y$ and $Y \rightarrow Z$ hold, then $X \rightarrow Z$ also holds.

Moreover, the two following rules are sound:

F_4 (union). If $X \rightarrow Y$ and $X \rightarrow Z$ hold, then $X \rightarrow YZ$ also holds

F_5 (decomposition). If $X \rightarrow YZ$ holds, then $X \rightarrow Y$ also holds

The *closure* X_F^+ of a set X of attributes with respect to a set F of FDs is defined as $X_F^+ = \{A \mid X \rightarrow A \in F^+\}$. It is easy to show that an FD $X \rightarrow Y$ is in F^+ if and only if $Y \subseteq X_F^+$. As a consequence, the implication problem can be solved by first computing X_F^+ and then checking whether Y is contained in X_F^+. This is a good solution, since X_F^+ can be computed very efficiently, as follows (and originally shown in Bernstein, 1976). A first, simple method is used by the following algorithm.

ALGORITHM 1. CLOSURE(X, F).

INPUT: a set F of FDs (over a scheme U)
and a set of attributes $X \subseteq U$

OUTPUT: the closure of X

```
METHOD: CLOSURE := X;
        OLDCLOSURE := { };
        WHILE CLOSURE ≠ OLDCLOSURE
        DO BEGIN
                OLDCLOSURE := CLOSURE;
                FOR ALL V → W ∈ F
                DO IF V ⊆ CLOSURE
                   THEN CLOSURE := CLOSURE ∪ W
           END;
        RETURN (CLOSURE)
```

In the worst case the algorithm executes the external loop once for each attribute and the internal once for each FD and so, if $|U| = p$ and $|F| = m$, the complexity of the algorithm is $O(m{*}p)$. The correctness of the algorithm is proved showing that X_F^+ contains and is contained in the set of attributes that is the value of the variable CLOSURE at the end of the execution of the algorithm. Both parts are proved by induction, one on the number of steps executed by the algorithm and the other one on the length of a derivation of $X \to X_F^+$ from X. At this point, the membership algorithm is immediate, and it also runs in time $O(m^*p)$: given F and $X \to Y$, it first computes X_F^+, and then checks whether $Y \subseteq X_F^+$.

A modified version of Algorithm 1 runs in time proportional to the global length of the input n. We omit its description, which can be found in Beeri and Bernstein (1979) and Maier (1983), but we will use the result in the following. The membership algorithm remains unchanged, but it also runs in linear time.

2.3. *Functional Dependencies and Null Values*

According to Lien (1982), a *functional dependency with nulls* (NFD) $X \to Y$ holds in a relation r over a scheme $R(U)$ (with $XY \subseteq U$) when, for each pair of X-total tuples $t_1, t_2 \in r$, if $t_1 \cdot X = t_2 \cdot X$, then $t_1 \cdot Y = t_2 \cdot Y$.

For null-free relations the definition of NFD reduces to that of FD and so it is a correct generalization of the concept. Moreover, it is coherent with the no-information interpretation. In fact, tuples with nulls in attributes in X cannot cause a violation of a dependency $X \to Y$: the nulls mean that no-information is available about those attributes. On the other hand, two X-total tuples, t_1, t_2, such that $t_1 \cdot X = t_2 \cdot X$ and t_2 is A-total while t_1 is not, violate a dependency $X \to Y$ with $A \in Y$: the first tuple indicates that no-information is available about the value for A associated with $t_1 \cdot X$, while the second indicates that the value for A associated with $t_2 \cdot X = t_1 \cdot X$ does exist, and this violates the natural definition of functional dependency that if the values for X are the same for two tuples, both tuples must contain the same information for the attributes in Y. It should be noted that this definition of satisfaction refers to properties of our knowledge of the real world, while it is generally stated that integrity constraints express properties of the real world itself. On the other hand, for databases without null values it is implicitly assumed that the real world is represented faithfully, and so that our knowledge is complete, and coincide with the real world; but, even in this case, databases are approximations of the real world, and our knowledge is far from being complete, and dependencies are compared with the available data, that is, our knowledge.

With respect to inference rules, it is immediate to prove that reflexivity,

A	B	C
a_1	$\varnothing$	c_1
a_1	$\varnothing$	c_2

FIGURE 2

augmentation, union, and decomposition are sound rules for NFDs also, while transitivity is not, as shown by the counterexample relation in Fig. 2, which satisfies both $A \rightarrow B$ and $B \rightarrow C$ but does not satisfy $A \rightarrow C$. It is clear from the example that the unsoundness of the rule is caused by the presence of nulls in the attribute(s) Y (B in the example) which implement the transitivity.

However, the other four rules are complete for the derivation of NFDs.

THEOREM 1 (Lien, 1982). *F_1, F_2, F_4, F_5 form a sound and complete set of inference rules for NFDs.*

As for the classical FDs, the concept of closure of a set of attributes with respect to a set of NFDs can be defined and used as the basis for the membership algorithm. We will indicate the closure of X with respect to a set N of NFDs with X_N^+. On the other hand, the closure cannot be computed by means of Algorithm 1, because the transitivity rule is not sound, and the attributes added to the initial value of the variable CLOSURE cannot be used to add further attributes. The algorithm can be modified by replacing the current value of the variable CLOSURE in the comparison in the IF statement with its initial value, X. As a consequence, each NFD can be used at most once to add attributes to CLOSURE, and so the external loop can be eliminated, and the algorithm runs in time $O(m)$, proportional to the number of NFDs.

ALGORITHM 2. NCLOSURE(X, N).

INPUT: a set N of NFDs (over a scheme U)
and a set of attributes $X \subseteq U$

OUTPUT: the closure of X

METHOD: CLOSURE := X;
FOR ALL $V \rightarrow W \in N$
DO IF $V \subseteq X$ THEN CLOSURE := CLOSURE $\cup$ W
RETURN (CLOSURE)

THEOREM 2. *Algorithm 2 correctly computes the closure X_N^+ of X.*

Proof. We show that the final value of the variable CLOSURE, indicated with CLOSURE*, is equal to X_N^+:

1. CLOSURE* $\subseteq X_N^+$. Let $A \in$ CLOSURE*; if $A \in X$, A is trivially in X_N^+; otherwise, there is an NFD $V \rightarrow W$ in N such that $V \subseteq X$ and $A \in W$; in this case, $X \rightarrow A$ can be obtained from N using the augmentation and decomposition rules, and so A is in X_N^+.

2. $X_N^+ \subseteq$ CLOSURE*. Let A be in X_N^+: this means that $X \to A$ is in N^+ and so (since the rules are complete) is derivable from N by means of the four rules above. We prove the theorem by induction on the length of the derivation, with the following inductive hypothesis: "if $Z \subseteq X$ and $n: Z \to Y$ is derivable in not more than s steps, then Y is contained in CLOSURE*" (we use a possibly non-singleton set Y in the proof because intermediate NFDs in the derivation need not have singletons as right-hand sides).

Basis: $s = 1$. n is in N and so, when n is processed by the algorithm (and this will happen) Y is added to CLOSURE.

Induction: $s > 1$ and the inductive hypothesis holds for derivations of length less than s. n is the last NFD in the derivation, and it is there because it is in N (and in this case we can argue as above) or because it is derived from other NFDs by means of an inference rule. So, we have four cases, one for each rule.

1. Reflexivity. If $Y \subseteq Z \subseteq X$, it is included in CLOSURE since the beginning.

2. Augmentation. There are an NFD g: $V \to W$, derived in less than s steps, and a set of attributes T such that $VT = Z$ and $WT = Y$; since $V \subseteq Z \subseteq X$, and g is derived in less than s steps, W is in CLOSURE* and, since $T \subseteq X$ (and so it is in CLOSURE since the beginning), $Y \subseteq$ CLOSURE*.

3. Union. There are two NFDs $Z \to V$ and $Z \to W$, both derivable in less than s steps, such that $VW = Y$; by the inductive hypothesis, since $Z \subseteq X$, both V and W are included in CLOSURE* and so is VW.

4. Decomposition. There is an NFD $Z \to W$, derivable in less than s steps, such that $Y \subseteq W$; by the inductive hypothesis, since $Z \subseteq X$, $W \subseteq$ CLOSURE* and so (since $T \subseteq W$) $Y \subseteq$ CLOSURE*. ▌

2.4. *Constraints On Null Values*

We have already discussed in the Introduction the importance of constraints on null values and exemplified the situations in which they can be useful. As a consequence, we present here only the definitions, without any further discussion.

A *null-free subscheme* (NFS) is a constraint that requires that a certain subset U_S of a relation scheme U does not contain nulls. Without loss of generality, we can assume that on each relation scheme a single NFS is defined. We indicate it in the relation scheme, specifying, besides the relation name and the global set of attributes U, the subset U_S of U which must be null-free: $R(U; U_S)$. In the first example presented in the introduction, we would have $U = \{\text{SSNo, category, age, salary}\}$ and $U_S = \{\text{SSNo}\}$.

For NFSs alone, it is not meaningful to study derivation and implication, since the results are really trivial.

An *existence constraint* (EC) is a statement $e\colon X \vdash\!\!- Y$ (read X *requires* Y), where X, Y are sets of attributes. $X \vdash\!\!- Y$ holds in a relation r over a scheme $R(U)$ (with $XY \subseteq U$) if each X-total tuple $t \in r$ is also Y-total. If $Y = \{\ \}$, then the EC is assumed satisfied. ECs generalize NFSs, since any NFS U_S can be expressed by means of the EC $\{\ \} \vdash\!\!- U_S$. The second example in the introduction can be modelled by means of the EC: manager's first name $\vdash\!\!-$ manager's last name.

The study of inference rules for ECs leads to an interesting result (argued, without proof, in Maier, 1980): the rules of reflexivity, augmentation, transitivity, form a sound and complete set of inference rules for ECs.

THEOREM 3. *The rules*

E_1 (*reflexivity*). *If* $Y \subseteq X$, *then* $X \vdash\!\!- Y$ *holds*

E_2 (*augmentation*). *If* $X \vdash\!\!- Y$ *holds, then* $XZ \vdash\!\!- YZ$ *also holds*

E_3 (*transitivity*). *If* $X \vdash\!\!-$ *and* $Y \vdash\!\!- Z$ *hold, then* $X \vdash\!\!- Z$ *also holds*

form a sound and complete set of inference rules for the derivation of ECs.

Proof. 1. The soundness of the rules derives immediately from the definition of EC and its proof is left to the reader.

2. The proof of completeness is absolutely analogous to that of the rules for FDs without nulls (see, e.g., Ullman, 1982). The only differences are that a new concept, called the closure X_E^+ of a set of attributes X with respect to a set E of ECs, is used instead of the closure with respect to FDs,

$$X_E^+ = \{A \mid X \vdash\!\!- A \text{ is derivable from } E \text{ by means of } E_1 - E_3.\}$$

and that the counterexample relation has only one tuple, as

X_E^+	$U - X_E^+$
$1\,1 \cdots 1$	$\varnothing\,\varnothing \cdots \varnothing$

∎

Theorem 3 has the important consequence that most of the theory developed for FDs (including closure and membership algorithms) can be extended to ECs with no further effort. In the following the closure algorithm that, given a set E of ECs and a set of attributes X, returns the closure X_E^+ of X w.r.t. E, will be indicated with ECCLOSURE(X, E).

A *disjunctive existence constraint* (DEC, Maier, 1980) is a statement

$d: X \vdash S$, where X is a set of attributes and $S = \{Y_1, Y_2, ..., Y_n\}$ is a set of sets of attributes. $X \vdash \{Y_1, Y_2, ..., Y_n\}$ hold in a relation r over a scheme $R(U)$ (with $XY_1 Y_2 \cdots Y_n \subseteq U$) if for each X-total tuple $t \in r$, there is an $i \in \{1, 2, ..., n\}$ such that t is Y_i-total. If $n = 0$, the DEC is assumed not satisfied (this convention differs from that adopted in Goldstein, 1981, but seems more reasonable if we consider a definition based on logic), while if any of the Y_i is the empty set $\{\ \}$, the DEC is assumed satisfied. The third example in the introduction can be modelled by means of the DEC

last name $\vdash$ {SSNo, First name birth date birth place}.

Goldstein (1981) showed that there is a sound and complete set of inference rules for DECs. We present a slightly modified version of it:

D_1. If $Y \subseteq X$, then $X \vdash \{Y\}$ holds

D_2. If $X \vdash \{Y_1, Y_2, ..., Y_n\}$ holds, then, for any Z, $X \vdash \{Y_1, Y_2, ..., Y_n, Z\}$ also holds

D_3. If $X \vdash \{Y_1, Y_2, ..., Y_m\}$ and $X \vdash \{Z_1, Z_2, ..., Z_n\}$ hold, then $X \vdash \{Y_1 Z_1, ..., Y_1 Z_n, ..., Y_m Z_1, ..., Y_m Z_n\}$ also holds;

D_4. If $X \vdash \{Y_1, Y_2, ..., Y_m\}$ and, for some i, $Y_i \vdash \{Z_1, Z_2, ..., Z_n\}$ hold, then $X \vdash \{Y_1, ..., Y_{i-1}, Z_1, ..., Z_n, Y_{i+1}, ..., Y_m\}$ also holds.

Given a set D of DECs over a scheme $R(U)$ and a set of attributes $X \subseteq U$, let D^+ be the closure of D (i.e., the set of all DECs implied by D) and call D_X the set of DECs in D^+ whose left hand side is X. The *closure* of a set of attributes X with respect to D, indicated with X_D^+, is a set of subsets of U such that:

1. $\{X \vdash X_D^+\}$ and D_X are equivalent;

2. there is no other $\{X \vdash S\}$ equivalent to D_X such that S contains less subsets of U than X_D^+.

It is proved in Goldstein (1981) that the closure X_D^+ of X is unique. Unfortunately, only exponential algorithms are known for its computation and, as a consequence, for the membership problem. To confirm this result, in the remainder of this section, we prove that the implication problem for DECs is co-NP-complete, and so, unless $P = \text{co-}NP$, there is no polynomial time algorithm for its solution. We use a variant of the following result about co-NP-completeness of implication of propositional sentences, which derives immediately from NP-completeness of the satisfiability problem (Cook, 1971; Garey and Johnson, 1979).

LEMMA 1. *If S and s are propositional sentences in conjunctive normal form, determining whether S implies s is co-NP-complete.* ∎

A proof of the lemma is based on the fact that "a special case of the problem (namely, whether or not S implies $(p$ AND NOT$(p)))$ holds if and only if S is unsatisfiable" (Levesque, 1984).

COROLLARY 1. *If S is a propositional sentence in conjunctive normal form and s is a disjunction of literals, determining whether S implies s is co-NP-complete.*

Proof. Given a propositional variable p, solving two instances of our problem, namely "does S imply p?" and "does S imply NOT (p)?" we can determine whether S is unsatisfiable, and so our problem is at least as hard as a co-NP-complete one. On the other hand, it is a special case of the co-NP-complete problem in Lemma 1, and thus it cannot be harder than it. So, the problem is co-NP-complete. ▮

In the next theorem, we show that the implication problem for DECs is co-NP-complete, by means of a reduction from the implication problem in Corollary 1.

THEOREM 4. *The implication problem for DECs is co-NP-complete.*

Proof. 1. Completeness. We show a reduction from the co-NP-complete problem in Corollary 1 to the implication problem for DECs. Let $S = s_1$ AND $s_2 \cdots$ AND s_m, with $s_i = p_{i1}$ OR $p_{i2} \cdots$ OR p_{iM_i}, and $s = q_1$ OR $q_2 \cdots$ OR q_n, where all the p_{ij} and q_h are negated or non-negated literals from variables belonging to a given universe P. We build an instance of the implication problem for DECs, with a set D of DECs and a DEC d such that D implies d if and only if S implies s. Consider a universe of attributes U of the same cardinality as P, and establish a one to one correspondence between the two sets; now the set D is composed of a DEC d_i: $X_i \vdash \{A_{i1},..., A_{iK_i}\}$, for each conjunct s_i in S, where X_i is the set of the attributes corresponding to the negated variables in s_i, and the A_{ik} are the attributes corresponding to the non-negated variables in s_i; the DEC d is analogously defined from s.

For any given truth assignment T to the variables in P, consider a one tuple relation $r = \{t\}$, where $t \cdot A_j = \varnothing$ if and only if T assigns the value *false* to p_j (the other values in t are irrelevant). It is easy to show that, for any i, r satisfies d_i if and only if s_i assumes the value *true* under T: if r satisfies d_i, at least one of the following conditions is satisfied: (a) the value for at least one of the attributes in X_i is $\varnothing$; (b) the value for at least one of the attributes A_{ik} is not null. In either case the corresponding literal assumes the value *true*, and so s_j assumes the value *true*; the converse is analogous.

It is clear that the transformation from S, s to D, d can be performed in polynomial time. To complete the proof, we show that D implies d if and only if S implies s; actually, we show that D does not imply d if and only if S does not imply s. If S does not imply s, then there is a truth assignment which makes S *true* and s *false*; in such a case, r satisfies all the DECs in D and does not satisfy d. If D does not imply d, there is a counterexample relation r satisfying D and violating d; by the definition of DEC, the violation can be reconducted to a single tuple t; if, again, we consider the truth assignment T corresponding to t, we have that T makes S *true* and s *false*.

2. Membership in co-*NP*: by means of a reduction very similar to the one in the previous part, it is possible to reduce the implication problem for DECs to the problem in Lemma 1 which is in co-*NP*. ∎

3.0. Functional Dependencies and Null-free Subschemes

In this section we deal with the implication problem for NFDs in presence of NFSs. First of all we present a sound and complete set of inference rules and then show how it can be used to define an efficient membership algorithm. The set of rules was already presented in Atzeni and Morfuni (1984a).

In Section 2.3, presenting the inference rules for NFDs, we noted that the rule of transitivity is not sound because of the possible presence of nulls in the attributes that implement the transitivity. On the other hand, no null may appear in the attributes in U_S in any relation r over a relation scheme $R(U; U_S)$: so, if all the attributes in the middle term of the transitivity are in U_S, then transitivity holds.

THEOREM 5. *The following rule is sound:*

F'_3 *(null-transitivity). If $X \to Y$ and $Y \to Z$ hold and $Y - X \subseteq U_S$ then $X \to Z$ also holds.*

Proof. See Atzeni and Morfuni (1984a). ∎

Note that the augmentation rule is now redundant since it is derivable from the rules F_1, F'_3, F_4. Moreover, if $U_S = \{\ \}$, i.e., there is no restriction on the presence of nulls, F'_3 is redundant, since it is derivable from F_2, F_5. On the other hand, if $U_S = U$, i.e., no nulls are allowed, F'_3 reduces to the classical rule F_3.

THEOREM 6. *The rules F_1, F'_3, F_4, F_5 form a sound and complete set of inference rules for NFDs in relations with null-free subschemes.*

Proof. See Atzeni and Morfuni (1984a). ∎

Again, a concept of closure is introduced: the *closure* X_N^+ of a set of attributes X with respect to a set N of NFDs is the set of attributes defined as follows:

$$X_N^+ = \{A \mid X \to A \text{ is derivable from } N \text{ by means of the inference rules } F_1, F_3', F_4, F_5\}.$$

It is immediate from the rules of union and decomposition that $X \to Y$ is derivable from N by means of the rules if and only if $Y \subseteq X_N^+$. Moreover, $X \subseteq X_N^+$. As in the classical theory of FDs for X_F^+, we have $X_N^+ = \{A \mid X \to A \in N^+\}$. As a consequence, the membership algorithm is once again based on a closure algorithm, which is a modified version of Algorithm 1. The algorithm combines Algorithm 1 (applying rule F_3', that is, transitivity within the NFS) and Algorithm 2.

ALGORITHM 3. NFSCLOSURE(X, N).

INPUT: a set N of NFDs (over a scheme U with a null-free subscheme U_S) and a set of attributes $X \subseteq U$.

OUTPUT: the closure of X

METHOD:
```
CLOSURE := X;
OLDCLOSURE := { };
WHILE CLOSURE ≠ OLDCLOSURE
DO BEGIN
        OLDCLOSURE := CLOSURE;
        FOR ALL V → W ∈ N
        DO IF V ⊆ CLOSURE ∩ U_S
            THEN CLOSURE := CLOSURE ∪ W
    END;
FOR ALL V → W ∈ N
DO IF V ⊆ X THEN CLOSURE := CLOSURE ∪ W
RETURN (CLOSURE)
```

The proof of correctness of Algorithm 3 is similar to those of Algorithms 1 and 2, and is therefore omitted.

4.0. Functional Dependencies and Existence Constraints

In this section we develop the theory of functional dependencies when the presence of null values is controlled by means of existence constraints (ECs). So, we have a relation scheme $R(U)$ and a set of constraints $I = N \cup E$ on it, where N is a set of NFDs and E a set of ECs.

In the following, we show how, adding a new joint inference rule to the union of rules for NFDs and rules for ECs, we obtain a sound and complete set of inference rules for the joint class. However, as an intermediate step in the derivation of such set, we introduce an auxiliary constraint, called fictitious functional dependency (FFD). It will turn out to be useful in order to prove the completeness of the set of rules; furthermore, in the next section, when we deal with DECs, this constraint will become necessary to the derivation of a finite and complete set of inference rules.

The rules presented in Sections 2.3 and 2.4 are obviously sound also for the joint class, but they do not form a complete system. For instance, given the set of constraints

$$I = \{X \rightarrow Y, Y \rightarrow A, X \vdash\!\!- Y\},$$

it is impossible to derive, by means of the aformentioned rules, the NFD $X \rightarrow A$; on the other hand, given any relation r satisfying the constraints in I, for any pair of X-total tuples $t_1, t_2 \in r$ such that $t_1 \cdot X = t_2 \cdot X$ we have that, for the EC $X \vdash\!\!- Y$, they are also Y-total and, for the NFD $X \rightarrow Y$, $t_1 \cdot Y = t_2 \cdot Y$; then, for the NFD $Y \rightarrow A$, $t_1 \cdot A = t_2 \cdot A$ and so r satisfies $X \rightarrow A$. This means that $X \rightarrow A$ is implied by I. Let us analyze the example. We have something that resembles the property of transitivity. In Section 2.3 we have shown that, in presence of null values, the transitivity rule is not sound, because of the possible presence of null values in the middle term. In this case we have the EC $X \vdash\!\!- Y$ which guarantees that, when a tuple is X-total (and this is the only case in which it can cause a violation of the NFD $X \rightarrow A$) it is also Y-total, i.e., null-free in the middle term. So, we have proved the correctness of the rule:

If $X \rightarrow Y$, $Y \rightarrow A$, $X \vdash\!\!- Y$ hold, then $X \rightarrow A$ also holds.

Unfortunately, it does not form, together with the other rules, a complete system. Again, given the set of constraints

$$I = \{X \rightarrow Y, Y \rightarrow A, A \vdash\!\!- Y\},$$

it is still impossible to derive the NFD $X \rightarrow A$, which instead could be easily proved to be implied by I. In this case, it is the EC $A \vdash\!\!- Y$ that guarantees, when needed, the absence of null values on the middle term Y. So, we can generalize the two examples and come out with the following rule

If $X \rightarrow Y$, $Y \rightarrow A$, $XA \vdash\!\!- Y$ hold, then $X \rightarrow A$ also holds.

Again, the system is not yet complete: given the set of constraints

$$I = \{X \rightarrow YW, Y \rightarrow Z, WZ \rightarrow A, XA \vdash\!\!- YWZ\},$$

	A	B	C	D
r_1	a_1	b_1	$\varnothing$	d_1
	a_1	b_2	$\varnothing$	d_2
	a_2	b_1	c_2	d_1
	a_2	b_1	c_2	d_2
r_2	a_1	$\varnothing$	c_1	d_1
	a_1	$\varnothing$	c_2	d_2
	a_2	b_1	c_1	d_1
r_3	a_1	b_1	c_1	d_1
	a_1	$\varnothing$	c_2	d_1

FIGURE 3

the NFD $X \to A$, though non-derivable by means of the rules, is implied by I, as it can be proved by contradiction.

In the last example we have a transitivity that is performed in various steps, which, again, refer to null-free sets of attributes. In order to deal with cases as general as this, we introduce a new type of constraint, called fictitious functional dependency, which differs from the classical FD because it refers to tuples that are total on given sets of attributes, as suggested by the examples. A *fictitious functional dependency* (FFD) is a statement $X \to^Z Y$ (with $X \subseteq Z$). It holds in a relation r over a scheme $R(U)$ (with $YZ \subseteq U$) if for each $t_1, t_2 \in r$, if t_1 is Z-total and $t_1 \cdot X = t_2 \cdot X$, then $t_1 \cdot Y = t_2 \cdot Y$. Let us present an example. Given the relation scheme $R(ABCD)$, relations r_1, r_2 in Fig. 3 satisfy the FFD $A \to^{AB} C$ (which is meaningful because $A \subseteq AB$), while relation r_3 does not.

The satisfaction of $A \to^{AB} C$ in r_1 is equivalent to the satisfaction of the NFD $A \to C$, since all the tuples are AB-total. $A \to^{AB} C$ holds in r_2, because the only AB-total tuple is the last one and no other tuple agrees with it on the attribute A. In r_3, instead, the FFD is not satisfied, since the first tuple is AB-total and agrees with the second one on the attribute A, while they disagree on the attribute C. ▮

We are interested in FFDs only as a means for the derivation of NFDs, but, according to their definition, they could be considered as independent constraints. So, it is possible to study their derivation by means of inference rules. It comes out that the FFDs are so similar to the classical FDs that they have the same inference rules, apart from some technicalities, as stated by the following theorem, whose proof is therefore omitted.

THEOREM 7. *The following inference rules for FFDs are sound*:

FF_1 (*reflexivity*). *If* $Y \subseteq X \subseteq Z$, *then* $X \to^Z Y$ *holds.*

FF$_2$ (*augmentation*). *If $X \rightarrow^Z Y$ holds and $W \subseteq Z$, then $XW \rightarrow^Z YW$ also holds.*

FF$_3$ (*transitivity*). *If $X \rightarrow^W Y$ and $Y \rightarrow^W Z$ hold, then $W \rightarrow^W Z$ also holds.*

Now we introduce the rules which allow the derivation of FFDs from NFDs and viceversa, in presence of ECs.

First of all, it is immediate from the definition that each FFD $X \rightarrow^Z Y$ is strictly weaker than the NFD $X \rightarrow Y$. So the following rule for the joint class of NFDs, FFDs, ECs is sound:

J$_1$. If $X \rightarrow Y$ holds and $X \subseteq Z$, then $X \rightarrow^Z Y$ also holds.

The following theorem introduces and proves the soundness of the rule that allows the derivation of new NFDs, according to what is suggested by the examples.

THEOREM 8. *The following rule is sound*:

J$_2$. *If $X \rightarrow^Z A$ and $XA \vdash Z$ hold, then $X \rightarrow A$ also holds.*

Proof. We proceed by contradiction. Suppose that there exists a relation r satisfying $X \rightarrow^Z A$ and $XA \vdash Z$ and not satisfying $X \rightarrow A$. Then, there must be two X-total tuples $t_1, t_2 \in r$ such that $t_1 \cdot X = t_2 \cdot X$ and $t_1 \cdot A \neq t_2 \cdot A$. Thus, at least one of them is XA-total and so, for the EC $XA \vdash Z$, also Z-total; but this means that r does not satisfy $X \rightarrow^Z A$, against the hypothesis. ∎

It is important to note that rule J$_2$ applies only to FFDs (and so derives only NFDs) whose right-hand side is a singleton. It is easy to show that its generalization

If $X \rightarrow^Z Y$ and $XY \vdash Z$ hold, then $X \rightarrow Y$ also holds

is not sound.

From the joint rules (J$_1$, J$_2$) we have that $X \rightarrow^{XA} A$ and $X \rightarrow A$ are equivalent. So, since an NFD $X \rightarrow Y$ (with $Y = A_1 A_2 \cdots A_n$) is derivable if and only if $X \rightarrow A_1$, $X \rightarrow A_2, \ldots, X \rightarrow A_n$ are derivable, the derivation of $X \rightarrow Y$ is equivalent to the derivation of $X \rightarrow^{XA_1} A_1$, $X \rightarrow^{XA_2} A_2, \ldots$, $X \rightarrow^{XA_n} A_n$. Moreover, NFDs are special cases of FFDs: an FFD $X \rightarrow^{XA} B$ is an NFD if $A = B$, since the FFD $X \rightarrow^{XA} A$ is equivalent to the NFD $X \rightarrow A$.

The new rules can handle the examples given at the beginning of this section. Let us consider again the more general of them, which subsumes the others:

$$I = \{X \rightarrow YW, Y \rightarrow Z, WZ \rightarrow A, XA \vdash YWZ\}.$$

We can derive (by J_1) $X \to^{XYWZ} YW$, $Y \to^{XYWZ} Z$, $WZ \to^{XYWZ} A$, and (since the decomposition rule, as well as the union rule, holds for FFDs too) $X \to^{XYWZ} Y$, $X \to^{XYWZ} W$. Then (by FF_3) $X \to^{XYWZ} Z$, (by union) $X \to^{XYWZ} WZ$ and (by FF_3) $X \to^{XYWZ} A$. Finally, from $XA \vdash YWZ$ we have (by E_2) $XA \vdash XYWZ$ and (by J_2) $X \to A$.

The example shows the way in which, adapting the property of transitivity, FFDs allow the derivation of NFDs; the following three operations are applied repeatedly:

1. by means of rule J_1, the FFDs corresponding to the NFDs are derived;

2. by means of rules FF_1, FF_2, FF_3, all the FFDs are derived (possibly applying the property of transitivity);

3. by means of rule J_2, NFDs are derived as special cases of FFDs.

We can now state and prove the theorem that guarantees the completeness of the rules presented until now for the derivation of constraints of the joint class.

THEOREM 9. *The rules F_1, F_2, F_4, F_5, E_1, E_2, E_3, FF_1, FF_2, FF_3, J_1, J_2 form a sound and complete system for the derivation of NFDs and ECs.*

Proof. Soundness has already been proved. With regard to completeness, we proceed as usual, showing that for each constraint i non-derivable from a given set I by means of the rules there is a counterexample relation r satisfying all the constraints in I and not satisfying i.

Let $I = E \cup N$, where E is a set of ECs and N a set of NFDs. If e is an EC that cannot be derived from I by means of the rules, we can consider the same counterexample relation as in the proof of Theorem 3, which does not satisfy e and satisfies all the ECs in E and trivially satisfies all the NFDs, since it contains only one tuple.

If $n: X \to Y$ is an NFD that cannot be derived from I by means of the rules, then there must be an attribute $B \in Y$ such that $X \to B$ cannot be derived (otherwise $X \to Y$ would be derivable by means of the union rule). Now let $Z = (XB)_E^+$,

$$X_{FF}^+ = \{A \mid A \in Z \text{ and } X \to^Z A \text{ is derivable from } I \text{ by means of the rules}\}$$

and r be the two tuple relation:

r	X_{FF}^+	$Z - X_{FF}^+$	$U - Z$
	$1\,1 \cdots 1$	$1\,1 \cdots 1$	$\varnothing\varnothing \cdots \varnothing$
	$1\,1 \cdots 1$	$2\,2 \cdots 2$	$\varnothing\varnothing \cdots \varnothing$

1. r satisfies all the ECs in E. Let e: $V \vdash\!\!- W \in E$: if $V \not\subseteq (XB)_E^+$, then e is trivially satisfied; otherwise, $XB \vdash\!\!- V$ is derivable and, by transitivity, also $XB \vdash\!\!- W$ is derivable and then $W \subseteq (XB)_E^+$ and so e is satisfied.

2. r satisfies all the NFDs in N. Let n: $V \rightarrow W \in N$. If $V \not\subseteq X_{FF}^+$, n is trivially satisfied. If $V \subseteq X_{FF}^+$, $X \rightarrow^Z V$ is derivable. We proceed showing that for each $A \in W$, $V \rightarrow A$ is satisfied (and so, by the union rule, $V \rightarrow W$ is satisfied). If $A \notin Z$, $V \rightarrow A$ is satisfied. If $A \in Z = (XB)_E^+$, then from $V \rightarrow A$ and $V \subseteq X_{FF}^+ \subseteq Z$ (for J_1) $V \rightarrow^Z A$ is derivable and (for FF_3) $X \rightarrow^Z A$ is derivable and so $A \in X_{FF}^+$ and then $V \rightarrow A$ is satisfied.

3. r does not satisfy $X \rightarrow B$. For the definition of $Z = (XB)_E^+$, $B \in Z$. If $B \in X_{FF}^+$, $X \rightarrow^Z B$, $XB \vdash\!\!- Z$ and (for J_2) $X \rightarrow B$ would be derivable, against the hypothesis. So, $B \in Z - X_{FF}^+$ and the two tuples agree on X and disagree on B. ▮

Now, we introduce a new joint rule which, together with those already presented, forms a sound and complete set for the derivation of NFDs and ECs, without using FFDs. However, the proof of the completeness theorem makes use of FFDs, thus motivating their introduction. The rule is:

J_3. If $XY \rightarrow A$, $Z \rightarrow Y$ and $XA \vdash\!\!- Y$ holds, then $XZ \rightarrow A$ also holds.

THEOREM 10. *Rule J_3 is sound.*

Proof. Let r be a relation satisfying $XY \rightarrow A$, $Z \rightarrow Y$, and $XA \vdash\!\!- Y$; we show that it satisfies $XZ \rightarrow A$, that is, we show that for any pair of XZ-total tuples $t_1, t_2 \in r$ such that $t_1 \cdot XZ = t_2 \cdot XZ$ we have $t_1 \cdot A = t_2 \cdot A$. If this is not the case, at least one of them, say t_1, is XA-total and so, for the EC $XA \vdash\!\!- Y$, also Y-total. On the other hand, since the tuples are Z-total with $t_1 \cdot Z = t_2 \cdot Z$ and r satisfies the FD $Z \rightarrow Y$, it follows that t_2 is also Y-total and $t_1 \cdot Y = t_2 \cdot Y$. So, we have that the two tuples are XY-total and $t_1 \cdot XY = t_2 \cdot XY$; since r satisfies $XY \rightarrow A$, we have $t_1 \cdot A = t_2 \cdot A$. ▮

Let us consider again the examples presented at the beginning of this section. In the first two cases, $I = \{X \rightarrow Y, Y \rightarrow A, X \vdash\!\!- Y\}$ and $I = \{X \rightarrow Y, Y \rightarrow A, A \vdash\!\!- Y\}$, for the respective augmentation and decomposition rules we can derive the EC $XA \vdash\!\!- Y$, and the NFD $XY \rightarrow A$, and then apply rule J_3, with $Z = X$ (we obtain exactly $XY \rightarrow A$, $X \rightarrow Y$, $XA \vdash\!\!- Y$), to derive $X \rightarrow A$. In the last example

$$I = \{X \rightarrow YW, Y \rightarrow Z, WZ \rightarrow A, XA \vdash\!\!- YWZ\}$$

we have to apply J_3 twice. In fact, from $XWZ \rightarrow A$, $YW \rightarrow WZ$, and $XA \vdash\!\!- WZ$, we derive $XYW \rightarrow A$; and then from $XYW \rightarrow A$, $X \rightarrow YW$, and $XA \vdash\!\!- YW$, we obtain $X \rightarrow A$.

THEOREM 11. *The rules F_1, F_2, F_4, F_5, E_1, E_2, E_3, J_3 form a sound and complete set of inference rules for the joint class of NFDs and ECs.*

Proof. Soundness has already been proved. Let $n: X \to A$ be an NFD in $I^+ = (N \cup E)^+$. We know, from Theorem 9, that n is derivable from I by means of F_1, F_2, F_4, F_5, E_1, E_2, E_3, FF_1, FF_2, FF_3, J_2, J_2. If no FF-rule is used in the derivation, n is derivable by means of F-rules only and so we have proved the thesis.

Otherwise, the derivation of $X \to A$ requires the derivation of the FFD $X \to^Y A$, for some Y with $XA \vdash Y$, and, then, the use of rule J_2. With respect to FFDs, we can restrict our attention to those of the form $X \to^Y Z$, since rules FF_1–FF_3 allow only the derivation of FFDs referring to the same set Y of attributes. Without loss of generality, we can assume that Y is as large as possible, that is $Y = (XA)^+_E$. Now, n can be obtained through a derivation organized in three parts, as follows:

1. Derivation from N, by means of J_1, of all the useful FFDs. So, for each $V \to Z \in N$ such that $V \subseteq Y$, $V \to^Y Z \cap Y$ (the other attributes in Z cannot give any contribution to the derivation, since they would never appear in a left-hand side).

2. Derivation of the FFD $X \to^Y A$. For the above discussion, this is possible, and we can derive it by first deriving the FFD $X \to^Y W$ with the largest W, and then from it obtain $X \to^Y A$. Since FFDs have exactly the same behavior as FDs, W is the closure of X with respect to the set of FFDs, and can be computed by an algorithm similar to Algorithm 1. So we can derive the FFDs $X \to^Y X_i$, for $i = 0, 1, \ldots, p$, with $X_0 = X$, the FFD $V_i \to^Y W_i$ derived in step 1 (for $i = 1, \ldots, p$), $V_{i+1} \subseteq X_i$, $X_{i+1} = X_i W_{i+1}$, and $X_p = W$.

3. Derivation of $X \to A$ from $X \to^Y A$ and $XA \vdash Y$, by means of J_2.

Now, we show how $X \to A$ can be derived by means of the F-rules, E-rules, and rule J_3, that is, with no derivation of FFDs. We prove that, for every $i = 0, 1, \ldots, p$, $X_i \to A$ is derivable by means of the rules; as a special case, $i = 0$, we will derive the thesis. We proceed by induction on $j = p - i$.

Basis: $j = 0$ (i.e., $i = p$). Since $A \in X_p$, $X_p \to A$ derives by reflexivity.

Induction: $j > 0$ (i.e., $i < p$). By the inductive hypothesis, we know that the thesis holds for $j - 1$, and so $X_{i+1} \to A$ is derivable by means of the rules. On the other hand, X_{i+1} is the union of X_i and W_{i+1}, with $V_{i+1} \to W_{i+1}$ in N^+, W_{i+1} contained in Y (with $XA \vdash Y$ in E^+). So, we have

1. $X_i W_{i+1} \to A$ derivable by means of the rules
2. $V_{i+1} \to W_{i+1} \in N^+$
3. $XA \vdash W_{i+1} \in E^+$.

Thus, using rule J_3, we can derive $X_i V_{i+1} \to A$, which, since $V_{i+1} \subseteq X_i$, reduces to $X_i \to A$. ▌

It is interesting to note that the interaction between NFDs and ECs actually exists only with respect to NFDs: given a set $I = N \cup E$, the ECs in I^+ are exactly those in E^+. This is due to the fact that the predicate defining ECs refers to one tuple only, while that of NFDs refers to a pair of tuples. As a consequence, the membership and closure algorithms for ECs are again those in Section 2.4. With respect to NFDs, the membership algorithm is still the same, but with a different closure algorithm, which takes into account the new rules. Essentially, the algorithm first computes, for each A in $U - X$, the closure $(XA)^+_E$ of XA with respect to the ECs in E, and then computes the closure of X with respect to the NFDs in N, allowing transitivity to be applied, when considering the attribute A, only on the attributes in $(XA)^+_E$. Without loss of generality, the algorithm assumes that all the NFDs in input have a singleton as their right-hand side.

ALGORITHM 4. NFECCLOSURE(X, N, E).

INPUT: a set N of NFDs and a set E of ECs over a scheme U and a set of attributes $X \subseteq U$

OUTPUT: the closure of X with respect to the set of NFDs implied by $N \cup E$.

```
METHOD: FOR ALL A ∈ U − X
        DO ECCL (A) := ECCLOSURE (XA, E);
        CLOSURE := X;
        OLDCLOSURE := { };
        WHILE CLOSURE ≠ OLDCLOSURE
        DO BEGIN
                OLDCLOSURE := CLOSURE;
                FOR ALL Y → A ∈ N
                DO IF Y ⊆ CLOSURE AND
                         Y ⊆ ECCL (A)
                   THEN CLOSURE := CLOSURE ∪ A
           END;
```

THEOREM 12. *Algorithm 4 correctly computes the closure X^+_N of X with respect to N, in the presence of E.*

Proof. We show that the final value of the variable CLOSURE, w.r.t. an input X, indicated with CLOSURE*(X), is equal to X^+_N.

1. CLOSURE*(X) $\subseteq X^+_N$. If A is added to CLOSURE, then A is in X^+_N. The proof is by induction on the number j of executions of the inner

loop, the inductive hypothesis being that, after the execution, the current value of the variable CLOSURE is contained in X_N^+.

Basis: $j=0$. A is in X and so in X_N^+, since $X \to A$ is derived by reflexivity.

Induction: $j>0$. For the inductive hypothesis, we can assume that before the execution of the IF statement, the value of the variable CLOSURE is contained in X_N^+ and equal to the value of OLDCLOSURE. If an attribute A is added to CLOSURE, there must be a set Y such that $Y \subseteq$ OLDCLOSURE, $Y \subseteq (XA)_E^+$ and $Y \to A \in N$. So, since $Y \subseteq X_N^+$, $X \to Y$ is derivable; furthermore, from $Y \to A$, by augmentation and decomposition, we can derive $XY \to A$. Then, applying J_3 (with $Z=X$), we can deduce $X \to A$ and, so, $A \in X_N^+$.

2. $X_N^+ \subseteq$ CLOSURE*(X): if A is in X_N^+ (and so $X \to A$ is derivable), then A is in CLOSURE*(X). The proof is by induction on the length p of the derivation of $X \to A$ (actually, the inductive hypothesis uses an NFD with left hand side $W \subseteq X$, for technical reasons, and right-hand side Y, because NFDs of this more general form may appear in the intermediate steps). Moreover, we indicate with CLOSURE$^{(j)}(X)$ the value of the variable CLOSURE after j iterations of the inner loop, with CLOSURE$^{(0)}(X)=X$ and CLOSURE$^{(n)}(X)=$ CLOSURE*(X). We show that, if $A \in X_N^+$, then there is a j such that $A \in$ CLOSURE$^{(j)}(X)$ (note that it is immaterial to know whether $j>n$ since, in this case, CLOSURE$^{(j)}(X)=$ CLOSURE$^{(n)}(X)$: in fact, the algorithm always terminates). We assume that all the derivations of ECs necessary for the proof are made independently.

Basis: $p=1$. $W \to Y$ either follows by reflexivity, and so $A \in$ CLOSURE$^{(0)}(X)$, or $W \to Y$ is in N and so Y is added to CLOSURE before the end of the first execution of the outer loop.

Induction: $p>1$. $W \to Y$ is either in N or follows from the previous NFDs by one of the F-rules or by rule J_3. If $W \to Y$ is in N, or follows by F_1, then we may reason as in the basis. If it follows by F_2, F_4, F_5, we can reason as in the proof of correctness of Algorithm 2 (Theorem 2), with minor changes.

If $W \to A$ follows by J_3, then there are two previously derived NFDs, $VZ \to A$ and $T \to Z$, such that $VT=W$, with the EC $VA \vdash\!\!- Z$ also available. Since $T \to Z$ has a derivation of less than p lines, by the inductive hypothesis we know that (running the algorithm with T in place of X) there exists a j such that all the attributes in Z are in CLOSURE$^{(j)}(T)$. On the other hand, since $T \subseteq W \subseteq X$, CLOSURE$^{(j)}(T)$ is contained in CLOSURE$^{(j)}(X)$ (the easy proof, by induction on j is omitted) and so $Z \subseteq$ CLOSURE$^{(j)}(X)$. Thus, since also V is contained in X, we have $VZ \subseteq$ CLOSURE$^{(j)}(X)$. Furthermore, since $VA \subseteq XA$, $(VA)_E^+ \subseteq (XA)_E^+$. Now, consider running the algorithm with VZ in place of X. By the inductive hypothesis there is some k such that CLOSURE$^{(k)}(VZ)$ contains A.

Since $VZ \subseteq \text{CLOSURE}^{(j)}(X)$, and $(VZA)_E^+ \subseteq (XA)_E^+$ (since $VA \subseteq XA$ and $Z \subseteq (XA)_E^+$), we deduce that $\text{CLOSURE}^{(j+k)}(X)$ contains A. ▮

The complexity of Algorithm 4 can be easily computed since we know that the classical CLOSURE algorithm can run in time proportional to the length of the input. So, if we indicate with m the cardinality of U, with p the sum of the lengths of the dependencies in N and with q the sum of the lengths of the dependencies in E, we have that the first step (which executes the algorithm ECCLOSURE at most m times) requres time proportional to $m*q$ and the second step (which is a slightly modified version of Algorithm 1), requires a time proportional to p; so the entire algorithm can be implemented to run in a time $O(m*q+p)$.

5.0. Functional Dependencies and Disjunctive Existence Constraints

In this section we are concerned with the interaction between NFDs and DECs. The main result that will be obtained is that, while it is possible to generalize Theorem 9 to the new situation by means of an extension of rule J_2, it is impossible to prove a result similar to that proved in Theorem 11. In fact, we prove that there exists no finite, complete set of inference rules for the class of constraints containing only NFDs and DECs. Actually, even the new rule J_2' replacing J_2 is not a "finite" rule, because it involves an unbound number of premises, and so we do not obtain a finite, complete axiomatization. This is a situation analogous to that of embedded multivalued dependencies, and of the joint class of FDs and inclusion dependencies, which do not have finite axiomatizations, while have unbound axiomatizations (Sagiv and Walecka, 1982; Parker and Parsaye-Ghomi, 1980; and Casanova, Fagin, and Papadimitriou, 1984). We do not present algorithms for the implication problem for DECs and NFDs, because we already know (from Theorem 4) that the implication problem for DECs alone is intractable.

The way in which DECs interact with NFDs in the implication of new NFDs is substantially analogous to that of ECs. As a consequence, we will omit details in the discussion, and the proofs, which are essentially extensions of proofs in Section 4.

The main difference between the interaction of NFDs and DECs and the interaction of NFDs and ECs is obviously related to their different definitions. Specifically, let us consider the inference rule J_2 for NFDs and ECs:

J_2. If $X \to^{Z} A$ and $CA \vdash Z$ hold, then $X \to A$ also holds.

If instead of the EC $XA \vdash Z$, we have a DEC $XA \vdash \{Y_1, \ldots, Y_n\}$, the absence of nulls on XA can in general guarantee the absence of nulls on one of the sets Y_i, and so in order to have the NFD $X \to A$, we need that all the FFDs $X \to^{Y_i} A$ are satisfied. The following theorem states a new rule for NFDs and DECs, which extends J_2.

THEOREM 13. *The rule*

J_2'. *If* $XA \vdash \{Y_1, ..., Y_p\}$ *and, for each* i, $X \to^{Y_i} A$ *hold, then* $X \to A$ *also holds*

is sound.

The next theorem states the completeness of the rules.

THEOREM 14. *The rules* F_1, F_2, F_4, F_5, D_1, D_2, D_3, D_4, FF_1, FF_2, FF_3, J_1, J_2' *form a sound and complete system for the derivation of NFDs and DECs.*

The next, and final, result that we want to prove is the non-existence of a finite, complete set of inference rules for the class of constraints containing only NFDs and DECs; this fact will confirm the importance of FFDs. The proof is organized as follows: we show that for any integer $k > 1$, there exists a set of NFDs and DECs closed under k-ary implication (i.e., with respect to inference rules with at most k premises) but not closed under implication; from a theorem of Casanova, Fagin, and Papadimitriou (1984), this implies that there can be no finite axiomatization for NFDs and DECs.

For any given $k > 1$, let $U = A_0 A_1 \cdots A_k$ the global set of attributes, and $I = N \cup D$, where

$$D = \{d\colon A_0 \vdash \{A_1, A_2, ..., A_{k-1}\}\}$$

and

$$N = \{A_0 \to A_1 \cdots A_{k-1}, A_1 \to A_k, ..., A_{k-1} \to A_k\}.$$

Furthermore, let

$$G = \{X \to Y \mid XY \subseteq U \text{ AND } (p_1 \text{ OR } p_2) \text{ AND } (p_3 \text{ OR } p_4)\}$$

where

$$\begin{aligned} &p_1 = A_k \notin Y - X \\ &p_2 = \exists i \in \{1, ..., k-1\} \quad \text{s.t.} \quad A_i \in X \\ &p_3 = \forall i \in \{0, ..., k-1\} \quad A_i \notin Y - X \\ &p_4 = A_0 \in X. \end{aligned}$$

Let us indicate with N^+ the closure of N in absence of any constraint on nulls (by Theorem 1 this set can be obtained from N by means of the rules F_1, F_2, F_4, F_5). We prove that G is exactly N^+. The proof is divided in two lemmas and one corollary.

LEMMA 2. $G \subseteq N^+$.

Proof. Let $n: X \to Y \in G$. We have to prove that $n \in N^+$. X and Y satisfy the condition $[(p_1 \text{ OR } p_2) \text{ AND } (p_3 \text{ OR } p_4)]$; so, we consider the four possible cases:

1. p_1 AND p_3: in this case $Y - X = \{\ \}$, and so $X \to Y$ is derivable by reflexivity.

2. p_1 AND p_4: $Y - X \subseteq A_1 \cdots A_{k-1}$ and $A_0 \in X$. In this case, since $A_0 \to A_1 \cdots A_{k-1}$ is in N, we can derive $X \to YX$ and then $X \to Y$ by augmentation (F_2) and decomposition (F_5).

3. p_2 AND p_3: $\exists i$ s.t. $A_i \in X$ and $Y - X \subseteq A_k$. If $Y - X = \{\ \}$, then $X \to Y$ is derivable by reflexivity; otherwise $(Y - X = A_k)$, since there is an A_i such that $A_i \in X$, and $A_i \to Y - X \in N$, we can derive, by augmentation and decomposition, $X \to Y$.

4. p_2 AND p_4: $\exists i$ s.t. $A_i \in X$ and $A_0 \in X$. In this case, from $A_0 \to A_1 \cdots A_{k-1}$ and $A_i \to A_k$ (which are in N), we can respectively derive, by augmentation and decomposition, $X \to A_0 A_1 \cdots A_{k-1}$ and $X \to A_k$; from them, by union, we can derive $X \to U$, and, finally, by decomposition, $X \to Y$. ∎

LEMMA 3. *G is closed under implication.*

Proof. We show that, applying rules F_1, F_2, F_4, F_5, to NFDs in G, we obtain NFDs also belonging to G.

1. Reflexivity. Let $n: X \to Y$ be the NFD derived by means of the rule. Then $Y - X = \{\ \}$ and so p_1 and p_3 are satisfied. Thus $n \in G$.

2. Augmentation. Let $n: X \to Y \in G$ and $Z \subseteq U$; then $n': XZ \to YZ$ is the derived NFD. Since $X \subseteq XZ$ and $YZ - XZ \subseteq Y - X$, it is immediate to verify that, if any of the conditions p_i holds for n, then it also holds for n'. So, if $n \in G$, then also $n' \in G$.

3. Union. Let $n_1: X \to Y_1$ and $n_2: X \to Y_2$ be in G; then $n: X \to Y_1 Y_2$ is the derived NFD. Since n_1, n_2 have the same left-hand side, either both satisfy condition p_2 or both do not satisfy it. So, we have two cases:

   a. p_2 holds for both n_1, n_2, and so also for n. So n satisfies p_1 OR p_2. With respect to p_3 OR p_4, we have again two possibilities.

1. p_4 holds for both n_1 and n_2, and so holds for n.
2. p_4 does not hold for n_1 nor for n_2; in this case both satisfy p_3, and so $Y_1 - X \subseteq A_k$ and $Y_2 - X \subseteq A_k$, from which we have $Y_1 Y_2 - X \subseteq A_k$, and so p_3 holds for n.

b. p_2 does not hold for n_1 nor n_2. Then, both satisfy p_1, $A_k \notin Y_1 - X$ and $A_k \notin Y_2 - X$, from which we have $A_k \notin Y_1 Y_2 - X$, and so n satisfies p_1. Thus n satisfies p_1 OR p_2; with respect to p_3 OR p_4, we can reason as above.

4. Decomposition. Let $n: X \rightarrow Y \in G$ and $n': X \rightarrow Z$ (with $Z \subseteq Y$) the NFD derived. In this case, for each of the conditions p_i, it is immediate that if p_i holds for n, then it also holds or n'. ▮

Corollary 2. $G = N^+$.

Proof. From the definitions of N and G, $N \subseteq G$; as a consequence, $N^+ \subseteq G^+$, and (since by Lemma 3, $G = G^+$) $N^+ \subseteq G$. On the other hand, by Lemma 2, $G \subseteq N^+$, and so the thesis $N^+ = G$ is proved. ▮

Now, we devote our attention to the influence of the DEC d on the set of NFDs implied by I.

Lemma 4. *The NFDs in $I^+ - N^+$ are exactly those of the form $A_0 \rightarrow Y$, with $A_k \in Y$.*

Proof. 1. If $A_k \in Y$, then $A_0 \rightarrow Y$ is in $I^+ - N^+$.

a. $A_0 \rightarrow Y \in I^+$, for every $Y \subseteq U$. First of all, $A_0 \rightarrow A_k \in I^+$: from $A_0 \vdash \{A_1, ..., A_{k-1}\}$, we derive $A_0 A_k \vdash \{A_0 A_1 A_k, ..., A_0 A_{k-1} A_k\}$, by augmentation, and from $A_0 \rightarrow A_1 \cdots A_{k-1}$, by decomposition, we obtain $A_0 \rightarrow A_i$, and then $A_0 \rightarrow^{A_0 A_i A_k} A_i$, for each $i \in \{1, ..., k-1\}$; analogously, for each $i \in \{1, ..., k-1\}$, from $A_i \rightarrow A_k$, we derive $A_i \rightarrow^{A_0 A_i A_k} A_k$; thus, for each i, by transitivity we derive $A_0 \rightarrow^{A_0 A_i A_k} A_k$ and, so, by rule J_2', $A_0 \rightarrow A_k$. Then, from $A_0 \rightarrow A_k$ and $A_0 \rightarrow A_1 \cdots A_{k-1}$, we obtain $A_0 \rightarrow A_1 \cdots A_k$ (by union), $A_0 \rightarrow U$ (by augmentation, since $U = A_0 A_1 \cdots A_k$), and, finally, $A_0 \rightarrow Y$ (by decomposition).

b. If $A_k \in Y$, then $A_0 \rightarrow Y \notin N^+$. Let us proceed by contradiction, assuming $A_k \in Y$ and $A_0 \rightarrow Y \in N^+$. In this case, from $A_0 \rightarrow Y$ we could derive $A_0 \rightarrow A_k$, by decomposition. But this is a contradiction, because, by Corollary 2, $A_0 \rightarrow A_k$ is not in N^+, since it does not satisfy condition (p_1 OR p_2).

2. If $n: X \rightarrow Y \in I^+ - N^+$, then n has the form $A_0 \rightarrow Y$ with $A_k \in Y$.

a. $X = A_0$. Since n does not belong to N^+, it does not satisfy the condition (p_1 OR p_2) AND (p_3 OR p_4). There are two cases:

1. (p_1 OR p_2) does not hold. Since p_2 is false, $X \subseteq A_0 A_k$. Since p_1 is false, A_k does not belong to X and so $X \subseteq A_0$. Since no NFD with empty left-hand side can be in I^+, $X = A_0$.
2. (p_3 OR p_4) does not hold. We show that, in this case, n does not belong to I^+, and so it does not need to be taken into consideration. Since p_3 is false, there exists an $i \in \{0, ..., k-1\}$ such that $A_i \in Y - X$, and, since p_4 is false, $A_0 \notin X$. Now, the following is a counterexample relation, which satisfies I and does not satisfy $X \to Y$ (note that A_0 and A_i can possibly coincide):

A_0	X	A_i	$(Y-A_i)-X$	$U-(A_0XY)$
$\varnothing$	$1 \cdots 1$	$\varnothing$	$1 \cdots 1$	$1 \cdots 1$
1	$1 \cdots 1$	1	$1 \cdots 1$	$1 \cdots 1$

b. $A_k \in Y$. Since $n \in I^+ - N^+$, $n \notin N^+$; for what we said in step 1.b, condition (p_1 OR p_2) does not hold. Then, if p_1 is violated, A_k is in Y. ▮

Lemma 5. *Let N' be a subset of N^+. If $N' \cup \{d\}$ implies, under k-ary implication, the NFD $A_0 \to YA_k$, then $N' \cup \{d\}$ implies, under k-ary implication, also the NFD $A_0 \to A_k$.*

Proof. It follows immediately from the consideration that the decomposition rule requires, for its application, only one premise. ▮

Lemma 6. *Let N' be a subset of N^+ with singletons as right hand sides of its NFDs. Then, for any $1 \leqslant i \leqslant k-1$, N' implies the FFD $A_0 \to^{A_0A_iA_k} A_k$ if and only if $A_0 \to A_i$ and $A_i \to A_k$ belong to N'.*

Proof. 1. (*If*) Since $A_0 \subseteq A_0A_iA_k$ and $A_i \subseteq A_0A_iA_k$, applying rule J_1 twice, we deduce $A_0 \to^{A_0A_iA_k} A_i$ and $A_i \to^{A_0A_iA_k} A_k$. By FF_3, it follows $A_0 \to^{A_0A_iA_k} A_k$.

2. (*Only if*) a. $A_0 \to A_i \in N'$. Assume it is not the case, then the following is a counterxample relation, which satisfies all the possible NFDs with a singleton as the right-hand side, except $A_0 \to A_i$ and $A_0 \to A_k$ (and so satisfies N'), and does not satisfy the FFD $A_0 \to^{A_0A_iA_k} A_k$. We have a contradiction:

A_0	$\cdots$	A_i	$\cdots$	A_k
1	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	1
1	$\varnothing \cdots \varnothing$	$\varnothing$	$\varnothing \cdots \varnothing$	$\varnothing$

b. $A_i \to A_k \in N'$. Again, assuming it is not the case, we have a counterexample relation, which satisfies all the NFDs, except those whose left-hand side is contained in $A_0 A_i$ and whose right-hand side contains A_k (and so satisfies N'), and does not satisfy $A_0 \to^{A_0 A_i A_k} A_k$:

A_0	$\cdots$	A_i	$\cdots$	A_k
1	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	1
1	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	$\varnothing$

∎

From the last lemma we derive the following two corollaries.

COROLLARY 3. *Let N' be subset of N^+ with singletons as right-hand sides of its NFDs. Then, N' implies the $k-1$ FFDs $A_0 \to^{A_0 A_j A_k} A_k$, for $1 \leqslant j \leqslant k-1$, if and only if the $2*(k-1)$ NFDs $A_0 \to A_i$, $A_i \to A_k$, $1 \leqslant i \leqslant k-1$, belong to N'.*

COROLLARY 4. *Let N' be a subset of N^+. Then, N' implies the $k-1$ FFDs $A_0 \to^{A_0 A_j A_k} A_k$, for $1 \leqslant j \leqslant k-1$, if and only if the k NFDs $A_0 \to A_1 \cdots A_{k-1}$, $A_i \to A_k$, $1 \leqslant i \leqslant k-1$, belong to $(N')^+$.*

LEMMA 7. *Let N' be a subset of N^+. Then,*

1. *$A_0 \to A_1 \cdots A_{k-1} \in (N')^+$, then for each $1 \leqslant i \leqslant k-1$ there exists a set Y_i such that $A_0 \to Y_i \in N'$ and $A_i \in Y_i$.*

2. *If $A_i \to A_k \in (N')^+$, then there exists a set Y such that $A_i \to Y \in N'$ and $A_k \in Y$.*

Proof. Both claims are proved by contradiction.

1. For each $1 \leqslant i \leqslant k-1$, consider the counterexample relation:

r	A_0	$\cdots$	A_i	$\cdots$	A_k
	1	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	$\varnothing$
	1	$\varnothing \cdots \varnothing$	$\varnothing$	$\varnothing \cdots \varnothing$	$\varnothing$

r satisfies all the NFDs in N' since N' contains no NFD $A_0 \to Y_i$, with $A_i \in Y_i$, by hypothesis. On the other hand, r does not satisfy $A_0 \to A_1 \cdots A_{k-1}$.

2. We can reason as in step 1, on the counterexample relation:

A_0	$\cdots$	A_i	$\cdots$	A_k
$\varnothing$	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	1
$\varnothing$	$\varnothing \cdots \varnothing$	1	$\varnothing \cdots \varnothing$	$\varnothing$

∎

COROLLARY 5. *Let N' be a subset of N^+. Then, N' implies the $k-1$ FFDs $A_0 \rightarrow^{A_0 A_j A_k} A_k$, for $1 \leqslant j \leqslant k-1$, if and only if:*

1. *For each $1 \leqslant i \leqslant k-1$, there exists a set Y_i such that $A_0 \rightarrow Y_i \in N'$ and $A_i \in Y_i$.*

2. *For each $1 \leqslant i \leqslant k-1$, there exists a set Y_i' such that $A_i \rightarrow Y_i' \in N'$ and $A_k \in Y_i'$.*

COROLLARY 6. *Let N' be a subset of N^+. If N' implies the $k-1$ FFDs $A_0 \rightarrow^{A_0 A_j A_k} A_k$, for $1 \leqslant j \leqslant k-1$, then N' has at least k dependencies.*

LEMMA 8. *Let I' be a generic set of NFDs and DECs on the set of attributes U'. Then, I' implies the NFD $X \rightarrow A$ (with $XA \subseteq U'$) if and only if for each set Z belonging to $(XA)_D^+$, I' implies the FFD $X \rightarrow^Z A$.*

Proof. 1. (*If*) From the soundness or rule J_2'.

2. (*Only if*) From the soundness of rule J_1 (since $Z \in (XA)_D^+$ implies $X \subseteq Z$, see Lemma 3.9 of Goldstein, 1981). ∎

THEOREM 15. *$N^+ \cup \{d\}^+$ is closed under k-ary implication.*

Proof. From Lemmas 4 and 5 it follows that it suffices to show that no subset N' of N^+ with cardinality not greater than $k-1$ implies, together with the DECs (which can be reconducted to d), $A_0 \rightarrow A_k$. Since $(A_0 A_k)_D^+ = \{A_0 A_1 A_k, ..., A_0 A_{k-1} A_k\}$, it remains to be proved, by Lemma 8, that no set of $k-1$ NFDs can imply the $k-1$ FFDs $A_0 \rightarrow^{A_0 A_i A_k} A_k$, $1 \leqslant i \leqslant k-1$. But this follows immediately from Corollary 6. ∎

As we said in the beginning of the section, a result of Casanova, Fagin, and Papadimitriou (1984), namely, Theorem 5, allows us to derive from Theorem 15 our final result.

THEOREM 16. *There can be no finite, complete set of inference rules for the class of constraints containing only NFDs and DECs.*

Acknowledgments

The authors would like to thank François Bancilhon, Carlo Batini, Ed Chan, Seymour Ginsburg, Rick Hull, Kent Laver, Alberto Mendelzon, Marina Moscarini, Stott Parker, and the two referees for their helpful comments.

Received February 14, 1985; revised December 2, 1985

References

Armstrong, W. W. (1974), Dependency structures of data base relationships, *IFIP Congr. Ser.* **580–583**.

Atzeni, P., and Morfuni, N. M. (1984a), Functional dependencies in relations with null values, *Inform. Process. Lett.* **18**, 233–238.

Atzeni, P., and Morfuni, N. M. (1984b), Functional dependencies and disjunctive existence constraints in database relations with null values, *in* "11th Colloq. Automata, Lang. Programming (ICALP)," Lecture Notes in Comput. Sci. Vol. 172, pp. 69–81, Springer-Verlag, Berlin/New York.

Beeri, C., and Bernstein, P. A. (1979), Computational problems related to the design of normal form relational schemes, *ACM Trans. Database Systems* **4**, 241–259.

Bernstein, P. A. (1976), Synthesizing third normal form relations from functional dependencies, *ACM Trans. Database Systems* **1**, 277–298

Casanova, M. A., Fagin, R., and Papadimitriou, C. H. (1984), Inclusion dependencies and their interaction with functional dependencies, *J. Comput. System Sci.* **28**, 29–59.

Codd, E. F. (1970), A relational model of data for large shared data banks, *Comm. ACM* **13**, 377–387.

Cook, S. A. (1971), The complexity of theorem proving procedures, *in* "Proceedings, 3rd Annu. ACM Sympos. on Theory of Computing," pp. 151–158, Assoc. Comput. Mach., New York.

Garey, M. R., and Johnson, D. S. (1979), "Computers and Intractability: A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco.

Goldstein, B. S. (1981), Constraints on null values in relational databases, *in* "Proc. 7th Internat. Conf. on Very Large Data Bases," pp. 101–110 IEEE Computer Society, Los Angeles.

Imielinski, T., and Lipski, W. (1983), Incomplete information and dependencies in relational databases, *in* "Proceedings, of ACM-SIGMOD Internat. Conf. on Management of Data," pp. 178–184, Assoc. Comput. Mach., New York.

Levesque, H. J. (1984), A logic of implicit and explicit belief, *in* "Proceedings, of the AAAI-84 Conerence," Austin, Texas; Fairchild Technical Report No. 653, Palo Alto, Calif.

Lien, Y. E. (1982), On the equivalence of database models, *J. Assoc. Comput. Mach.* **29**, 333–362.

Lipski, W. (1979), On semantic issues connected with incomplete information databases, *ACM Trans. Database Systems* **4**, 262–296.

Maier, D. (1980), "Discarding the Universal Instance Assumption: Preliminary Results," Technical Report 80-008, Dept. of Comput. Sci. SUNY at Stonybrook, N.Y., presented at the XP1 Workshop, Stonybrook, N.Y., June 1980.

Maier, D. (1983), "The Theory of Relational Databases," Comput. Sci. Press, Rockville, Md.

Parker, D. S., and Parsaye-Ghomi, K. (1980), Inference involving embedded multivalued dependencies and transitive dependencies, *in* "Proceedings, of ACM SIGMOND Internat. Conf. on Management of Data," pp. 52–57, Assoc. Comput. Mach., New York.

Sagiv, Y., and Walecka, S. F. (1982), Subset dependencies and a completeness result for a subclass of embedded multivalues dependencies, *J. Assoc. Comput. Mach.* **29**, 103–117.

Ullman, J. D. (1982), "Principles of Database Systems," 2nd ed., Comput. Sci. Press, Potomac, Md.

Vassiliou, Y. (1980), Functional dependencies and incomplete information, *in* "Proceedings, 6th Internat. Conf. on Very Large Data Bases," pp. 260–269, IEEE Comput. Soc., Los Angeles.

Zaniolo, C. (1984), Database relations with null values, *J. Comput. System Sci.* **28**, 142–166.